Overview of the fvDAS Experiment Setup and Runtime Scripts

Arlindo da Silva
Data Assimilation Office
dasilva@gsfc.nasa.gov

13 June 2000, Revised 13 December 2000

Outline

1. Introduction
2. Result of the Building Step
3. Core Architecture: Nested Executables
4. Main Applications/Utilities
5. Experiment Setup Utility (fvSETUP)
6. Demo: Running a Short Experiment
7. Producing Forecasts
8. Output Directory Layout and File Name Conventions
9. Pesto and Archiving Rules
10. Acquiring input data: The Acquire Utility

Introduction

- We distinguigh 3 basic steps from compiling to running fvDAS:

  Building step: The source code is compiled, with resulting executables, libraries, "headers", resource files, and scripts installed on a central directory. Example:

  /share/fvpsas/v1.2.beta.9/

  These binary releases will eventually be under CM.

  Experiment Setup: Using these pre-installed executables, an interactive utility (fvSETUP) creates an environment for carring out an experiment. A directory structure is created on a scratch area, e.g.,

  /scratch1/dasilva/v011b_b55/

  Experiment Execution: The experiment is carried out by means of short job segments, typically 1 day long. System output can be saved to mass-storaged or left on scratch area inspection and eventual archival.

- This presentation focus on experiment setup and execution.

Result of the Building Step

- FVROOT: directory where a pre-compiled version of the system has been installed.

- The following subdirectories are created under FVROOT:

  bin/ executables and scripts

  etc/ miscellaneous resource files

  lib/ libraries

  include/ compiled modules and include files

- The contents of lib/ and include/ are not necessary for running an experiment. They are include here to facilitate the development of utilities and customized applications.

- Example:

/share/fvpsas/v1.2.beta.9/

Core fvDAS Architecture: Nested Executables

```
type(time)      t    // current time
type(time)      dt   // analysis frequency
type(dyn_vect)  w_f  // state vector (u,v,h,q,dp)

// Initialize most things
...

// Main time loop...
t = t1
while ( t < t2 )

   call Atmos_Model ( t, t+dt, w_f )
   t = t + dt

   call Dyn_Put ( time, w_f )   // write state

   rc = system ( 'fvana' )      // run analysis

   call Dyn_Get ( time, w_f )   // retrieve state

endwhile
```

Main Applications/Utilities I

The following files are found under FVROOT/bin:

- Main executables:

fvpsas.x: General Circulation Model (fvCCM) executable with `system()` call for the analysis.

ana.x: Analysis application. It reads first guess and observations, producing post-analysis ODS (with O-F, O-A and QC marks) and analyzed state.

geos2fv.x: Utility for producing initial conditions from GEOS-2/3 fields.

blendq.x: Also used for preparing initial conditions. This utility blends moisture from AMIP and GEOS runs in the stratosphere.

- Main scripts:

  fvsetup: experiment setup utility (perl)

  fvpsas: Core DAS driver script (csh)

  fvana: Analysis driver script (perl)

  daotovs: DAOTOVS driver script (csh)

  fvarchive: mass-storage archiving driver (perl)

- Mass-storage Utilities:

  acquire: workhorse for input data acquisition (perl)

  pesto: archiving workhorse (csh)

  pasta: minor utility need by pesto (C)

- Post-processing utilities:

  dyn2prs.x: convert analysis GFIO files from eta to pressure coordinates.

  eta2prs.x: convert fvCCM prog and diag files from eta to pressure coordinates.

  rst_date: prints date on fvCCM dynamics restart file.

  fresolv.x: command-line interface to the File Resolver.

- Troubleshooting utilities:

  ainc.x: like ana.x but produces analysis increments on mandatory levels.

  obs.x: like ana.x but only produces post-analysis ODS (without O-A's).

  odsqc.x: reads an post-analysis ODS file and resets quality marks. Ideal for tuning QC of certain data types.

Experiment Setup Utility (fvSETUP)

- fvSETUP is an interactive `Perl` script which creates a runtime environment for running an experiment. Specifically:
  - Sets up FVHOME, the experiment home directory, which will contain restart, resource files and (temporarily) model output.
  - From user input, creates main job script and namelists
  - Copies resource files from FVROOT/etc to FVHOME/run
  - Copies initial conditions from user specified directory
  - Allow user to further customize namelists, resource files and main job script.
- A GUI version of fvSETUP (using perl/tk) is currently under development.

Demo: Running a Short Experiment

1. Logon to one of the NAS or NCCS machines where you plan to run your experiment on, *e.g.* `jimpf0`.

2. Select a pre-compiled fvDAS distribution, say the one under /share/fvpsas/v1.2.beta.8. Run fvSETUP from this distribution:

   ```
   jimpf0% /share/fvpsas/v1.2.beta.8/bin/fvsetup
   ```

3. Hit `RETURN` to accept the default for all questions, except for `Ending year-month-day?` which you should specify `19971215` for an 1 day assimilation experiment.

4. When the experiment setup is completed take a look at

   ```
   jimpf0% cd /scratch1/$user/v000_b55/run
   ```

   where `$user` is your user id.

5. To make your experiment run faster let's skip the computation of analysis increments with PSAS. You do this by editing the file ana.rc and changing one line:

   FROM:

   ```
   do_you_want_to_skip_PSAS: no   # yes or no
   ```

   TO:

   ```
   do_you_want_to_skip_PSAS: yes   # yes or no
   ```

6. Save the file and submit the job:

   ```
   jimpf0% qsub fvdas.j
   ```

   It should take about 15 minutes for this one day segment to complete.

7. By default the output will be saved to `helios1` at NAS, under your home directory, subdirectory `v000_b55/`.

Producing Forecasts

- During the experiment setup, the directory `$FVHOME/fcst` is created with the necessary scripts and resources for producing forecasts.

- By default, each assimilation job segment is 1 day long, and restart files are written out at the end of the segment, usually at `18Z`:

```
v000_b55.rst.lcv.19980101_18z.bin      (dynamics)
v000_b55.rst.phys.19980101_18z.bin     (physics)
v000_b55.rst.lsm.19980101_18z.bin      (land-surface)
```

- If specified during setup, additional restart files can be created and copied (staged) to `$FVHOME/fcst`.

- Users can always manually place restart files for one or more days under `$FVHOME/fcst`.

- For producing forecasts from each of these restarts, simply enter:

```
jimpf0% cd $FVHOME/fcst; qsub fvfcst.j
```

Main Resource Files

- Resource files contain configuration parameters which are fixed during an experiment.

- The following rc files are found under FVHOME/run by fvSETUP:

  ana.rc: analyzer, input ODS file names and selection of analysis levels; you can also choose to skip the the PSAS call.

  obs.rc: observer, data selection windows (red/yellow/ passive lists) and data thinning options.

  psas_qc.rc: statistical QC configuration

  daotovs.rc: DAO TOVS interactive retrievals

  tovs.rc: NESDIS TOVS bias coefficients (obsolete)

  psas.rc: PSAS configuration: conjugate gradient and error covariance parameters. **Note:** `superobs.rc` and `eagrid.rc` are currently the same file as `psas.rc`.

  ccmrun.namelist: main fvCCM namelist

Data Acquisition/Storage Resource Files

- Resource files that control the acquisition on input observational data:

  ana.acq: pre-analysis ODS files

  daotovs.acq: DAOTOVS radiances and colocated radiosonde data

- Resource files that control the archival of output data:

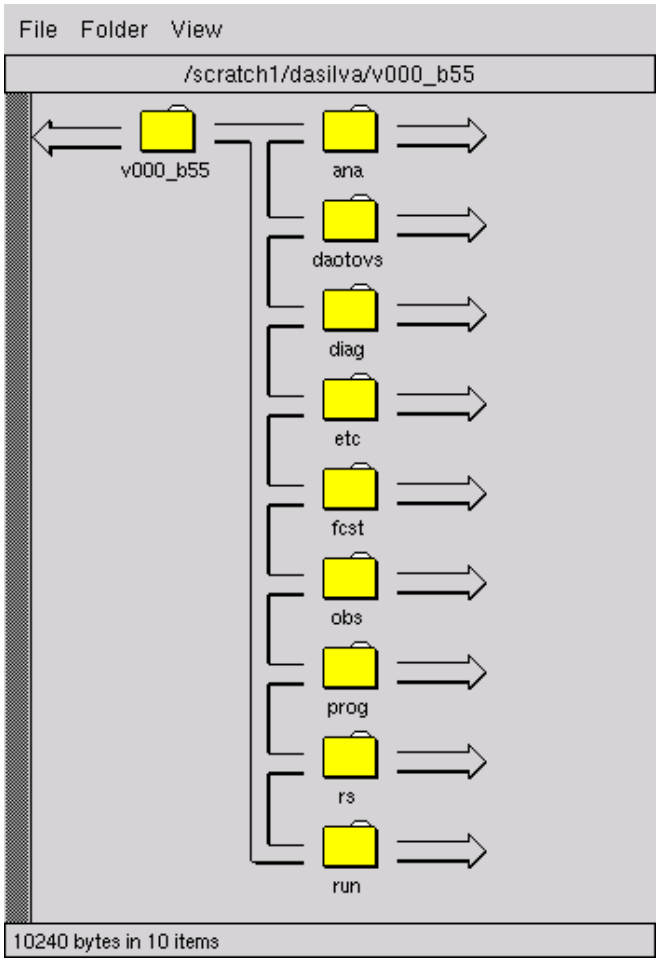
  silo.arc: archiving rules for transfering files from the temporary run directory to `$FVHOME`.

  mstorage.arc: archiving rules for transfering files from `$FVHOME` to mass storage.

- By default `silo.arc` and `mstorage.arc` are identical; by commenting out lines in `mstorage.arc` one can keep some of the output files on `$FVHOME`.

- However, if lines are commented out in `silo.arc` the corresponding output files will be forever lost.

FVHOME Directory Layout



FVHOME Directory Layout

ana/ analysis and background (first guess) files (GFIO)

daotovs/ iTOVS tuning and OMS files

diag/ diagnostic files (GrADS/IEEE)

etc/ job text output (log files)

fcst/ forecast script and staged restart files

obs/ post-analysis ODS files

prog/ fvCCM prognostic (refout) fields (GrADS/IEEE). Except for forecast fields, regular prognostic files are usually not produced as they have the same information as the background files.

rs/ fvCCM restart files (binary)

run/ jobs scripts, restarts, resource files

File Name Conventions

- fvDAS has been designed to allow a flexible, user defined file name convention.
- By default, it adopts the following file name convention:

$expid.$class.$coord.$time.$ftype

where the period delimited tokens are:

$expid: experiment ID, e.g., `v011b_b55`

$class: file class, e.g., `ana, prog, diag`

$coord: "vertical coordinate", e.g., `eta, lcv, prs`

$time: time tag, e.g., 19991203_12z

$ftype: file type, e.g., `txt, hdf, ods`

- Possible forms of the time tag:
  - 19991203, daily files
  - 19991203_18z, hourly files
  - 19991203_00z-19991207_18Z, files with a range of dates
  - 19991203_00z+19991203_06Z-19991207_18Z, forecast files starting on 0Z 3 Dec 1999 and valid from 6Z 3 Dec 1999 to 18Z 7 Dec 1999.
- Examples:
  - v011b_b55.ana.log.19990921.txt
  - v011b_b55.ana.eta.19990921.hdf
  - v011b_b55.bkg.eta.19990921.hdf
  - v011b_b55.ana.obs.19990921.ods
  - v011b_b55.diag.prs.19990921.ctl
  - v011b_b55.diag.prs.19990921.gmp
  - v011b_b55.diag.prs.19990921.grb

Managing fvDAS Output Files

- During execution, fvDAS writes its output files to a temporary directory. The name of this directory is recorded in the file:

  $FVHOME/.FVWORK

- When a job segment finishes, the relevant output files are moved to $FVHOME; the location of these files is specified in the text file:

  $FVHOME/run/silo.arc

- The `fvdas.j` script then submits a job to move these files to mass storage; the location of these files is specified in the text file:

  $FVHOME/run/mstorage.arc

  .

- The exact syntax of these *.arc files are discussed next.

(P)ut (E)xperiment in Mas (Sto)rage

```
NAME
     Pesto - (P)ut (E)xperiment in Mass [Sto]rage

SYNOPSIS
     pesto [-arc fname] [-d -ldir] [-h] [-n] [-r pestoroot] [-v]
           [-cp prog] [-rcp prog] [-rsh prog] [-m mode]

DESCRIPTION
     Pesto is a general purpose script for archiving experiments
     on mass storage. It requires the utility "pasta" and
     a resource file (see RESOURCE FILE below).

OPTIONS
 -arc fname    archiving resource file name (default: pesto.arc)
 -clean        delete files after transfer
 -d ldir       local directory for files being archived
 -h            displays this man page.
 -n            dry-run mode, does not copy anything, just print
 -r pestoroot  destination for archiving files (default: NONE)
 -v            verbose mode; default is real quiet
 -cp  prog     local  copy program  (default: cp)
 -rcp prog     remote copy program  (default: scp)
 -rsh prog     remote shell program (default: ssh)
 -m mode       destination files will have this mode (default: 0)
```

Sample Pesto Archiving Rules

```
#                   ---------------
#                   ANALYSIS FILES
#                   ---------------
#
${PESTOROOT}%s/ana/Y%y4/M%m2/%s.bkg.eta.%y4%m2%d2.hdf
${PESTOROOT}%s/ana/Y%y4/M%m2/%s.ana.eta.%y4%m2%d2.hdf
${PESTOROOT}%s/obs/Y%y4/M%m2/%s.ana.obs.%y4%m2%d2.ods
#
#                  -----------------
#                  PROGNOSTIC FILES
#                  -----------------
#
${PESTOROOT}%s/prog/Y%y4/M%m2/%s.prog.eta.%y4%m2%d2_%h2z-%y4%m2%d2_%h2z.ctl
${PESTOROOT}%s/prog/Y%y4/M%m2/%s.prog.eta.%y4%m2%d2_%h2z-%y4%m2%d2_%h2z.bin
#
#                  -----------------
#                  DIAGNOSTIC FILES
#                  -----------------
#
${PESTOROOT}%s/diag/Y%y4/M%m2/%s.diag.eta.%y4%m2%d2_%h2z-%y4%m2%d2_%h2z.ctl
${PESTOROOT}%s/diag/Y%y4/M%m2/%s.diag.eta.%y4%m2%d2_%h2z-%y4%m2%d2_%h2z.bin
#
...
```

Acquire: Input Data Acquisition

```
NAME
     acquire - Retrives files from mass storage with look-ahead capability

SYNOPSIS
     acquire [...options...]  bymd bhms ihms nstep

DESCRIPTION

     Acquire is a general purpose utility for retrieving files from
     mass storage with a look-ahead spooling capability. The full path
     names of the files to be retrieved are specified in a resource file
     (see RESOURCE FILE below) by means of GrADS-like templates. For
     example, a resource file could specify a file name of the form:

      gatun:/silo3/dao_ops/conv+tovs/ods/Y%y4/M%m2/r4ods.t%y4%m2%d2_%h2z

     The following parameters are required on input:

     bymd    beginning year-month-day, e.g., 19980101
     bhms    beginning hour-min-sec, e.g., 120000
     ihms    time step increment in hour-min-sec format
     nstep   number of timesteps for time looping.
```

Acquire Options

```
OPTIONS
     -cp            when files are to be preserved in spool directory
                    (see -p) a copy is made to the user destination directory
     -d path        destination (local) directory (default: ./)
     -e logfile     name of error log file. (default: no logging enabled)
     -f             forces remote copy whether the file exists locally or not
     -h             prints this page
     -lfn           use long file names in spool directory to
                    prevent file name conflicts
     -la lstep      Look ahead lstep time steps, that is, future data can be
                    pre-fetched and kept in spool directory (default: lstep=0)
     -p             preserve files in spool directory, simply making
                    a symlink() or copy to the local directory (see -cp);
     -rc fname      resource file name (default: acquire.rc)
     -s path        spool directory (default: /scratch1/dasilva/spool)
     -ssh           use ssh and scp (default: rsh and rcp)
     -strict        returns non-zero exit code if not all files
                    specified in the rc file are acquired
     -v             verbose mode (default is real quiet)
```

Sample Acquire Resource File

```
#
# Sample Acquire resource file.
#

#                   -------------
#                     ODS FILES
#                   -------------
#
gatun:/silo3/dao_ops/val/GEOS-2.9.0/conv+tovs/ods/rpk4atov/Y%y4/M%m2/r4aods.t%y4%m2%d2

#           -------------------------------
#           DAOTOVS LEVEL 1C RADIANCES, ETC
#           -------------------------------
#
helios1:/u/joiner/tovs/l1b/%y4%m2/l1c_new/amsua.nk.hdf.t%y4%m2%d2
helios1:/u/joiner/tovs/l1b/%y4%m2/l1c_new/hirs.nh.hdf.t%y4%m2%d2
helios1:/u/joiner/tovs/l1b/%y4%m2/l1c_new/hirs.nj.hdf.t%y4%m2%d2
helios1:/u/joiner/tovs/l1b/%y4%m2/l1c_new/hirs.nk.hdf.t%y4%m2%d2
helios1:/u/joiner/tovs/l1b/%y4%m2/l1c_new/msu.nj.hdf.t%y4%m2%d2
helios1:/u/joiner/tovs/l1b/%y4%m2/l1c_new/sonde_hirs.nh.hdf.t%y4%m2%d2
helios1:/u/joiner/tovs/l1b/%y4%m2/l1c_new/sonde_hirs.nj.hdf.t%y4%m2%d2
helios1:/u/joiner/tovs/l1b/%y4%m2/l1c_new/sonde_hirs.nk.hdf.t%y4%m2%d2
helios1:/u/joiner/tovs/l1b/%y4%m2/l1c_new/ssu.nh.hdf.t%y4%m2%d2
helios1:/u/joiner/tovs/l1b/%y4%m2/l1c_new/ssu.nj.hdf.t%y4%m2%d2
```